REMARKS

It is believed that this Amendment, in conjunction with the following remarks, place the application in immediate condition for allowance. Accordingly, entry of this Amendment under 37 C.F.R. § 1.116 and favorable consideration of the application are respectfully requested. Claims 20-24, 26, 32-35, 40, 49, 52, and 55 remain pending in this application.

Regarding the Objection to the Declaration

The Office Action objected to the Declaration because it makes reference to 37 C.F.R. § 1.56(a), rather than the entire rule, i.e., 37 CFR 1.56. In the previous Response, the Applicant argued that 37 C.F.R. § 1.56(a) implicitly incorporates the provisions set forth in 37 C.F.R. § 1.56(b) and 37 C.F.R. § 1.56(c). The Applicant acknowledges with appreciation that the Final Office Action states that this argument is persuasive with respect to 37 C.F.R. § 1.56(b) and 37 C.F.R. § 1.56(c).

The Final Office Action, states, however, that the analysis provided in the previous Response does not address the provisions of 37 C.F.R. § 1.56(d) and 37 C.F.R. § 1.56(e). These sections are reproduced below:

> (d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.
>
> (e) In any continuation-in-part application, the duty under this section includes the duty to disclose to the Office all information known to the person to be material to patentability, as defined in paragraph (b) of this section, which became available between the filing date of the prior application and the national or PCT international filing date of the continuation-in part application.

These sections are not relevant to the inventors' obligations under 37 C.F.R. § 1.56 in the present case. Namely, 37 C.F.R. § 1.56(d) is directed to "individuals *other than* the attorney, agent or inventor" (with emphasis added), and therefore has no bearing on the inventors' obligations under 37 C.F.R. § 1.56. (Note that it is the inventors' obligations that are at issue here, since the inventors are the individuals signing the Declaration.) 37 C.F.R. § 1.56(e) pertains only to continuation-in-part applications. Since, this application is not a continuation-in-part application, the provisions of 37 C.F.R. § 1.56(e) do not apply to the inventor's obligations under 37 C.F.R. § 1.56. Based on this analysis, the Applicant submits that the Declaration's reference to 37 C.F.R. § 1.56(a) effectively sets forth *all of the pertinent obligations* of the inventors that are contained in the entire rule, i.e., 37 C.F.R. § 1.56. The Declaration should be held to be compliant for this reason.

Alternatively, the Applicant submits that the declaration's deviation from more preferable language is minor and can be waived under MPEP § 602.03. The Patent Office is respectfully requested to exercise its authority and waive the declaration's reference to § 1.56(a), rather than § 1.56. This is considered appropriate because, as set forth in the expanded analysis above, the declaration implicitly complies with *ALL* of the pertinent provisions of 37 C.F.R. § 1.56.

Objection to the Drawings

The drawings were objected to under 37 C.F.R. 1.83(a) because the Office Action states that the drawings do not illustrate that the "request contains an identity of a desired locale" (as set forth in claims 57, 61, 65, 69). The Applicant points out that Fig. 1, in its original form, shows the submission of "Requests." Other figures, such as Fig. 6, show the use of a LocaleID parameter to convey locale information. However, to more directly respond to the Office Action's objection, Fig. 1 has been amended to include the

parenthetical label "(LocaleID)." This label is placed in positional association with the label "Requests." Support for this amendment can be found at least on page 25, lines 10-14 of the specification. In view of the above changes and remarks, the Patent Office is respectfully respected to remove the objection to the drawings.

35 U.S.C. § 103 Rejections

Claims 20-24, 26, 32-35, 49, 52, and 55 were rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of U.S. Patent No. 5,678,039 to Hinks et al. (referred to below as "Hinks") in view of "OpenWindows Developer's Guide: Xview Code Generator Programmer's Guide," by Sun Microsystems (referred to below as "Sun"), and further in view of U.S. Patent No. 6,802,059 to Lyapustina et al. (referred to below as "Lyapustina"). Claims 32-35 and 40 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hinks, Sun, Lyapustina, and U.S. Published Patent Application No. 2002/0107684 to Gao (referred to below as "Gao"). Claims 57-60 and 65-72 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hinks, Sun, Lyapustina, and U.S. Patent No. 6,308,212 to Besaw (referred to as "Besaw" below). Claims 61-64 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Hinks, Sun, Lyapustina, Gao, and Besaw. Applicant respectfully traverses each of these rejections for the reasons stated below.

Consider independent claim 20, reproduced below in full for the convenience of the Patent Office:

20. A method comprising:

compiling a computer-servable document written for a particular locale to extract and remove characters associated with any original locale-sensitive content, the compiling producing a compiled document with locale-independent elements;

storing the original locale-sensitive content;

substituting a function call in place of associated removed original locale-sensitive content in the compiled document; and

at runtime, retrieving the compiled document and populating the compiled document with a desired version of the original locale-sensitive content,

wherein the populating comprises executing the function call in the compiled document to obtain the desired version of the associated original locale-sensitive content and to insert the desired version of the associated original locale-sensitive content back into the compiled document.

The combination of Hinks, Sun, and Lyapustina was applied against this claim. By way of broadly-stated outline, the Office Action relies on Hinks for disclosing aspects of the compiling, storing, and retrieving operations of claim 1. The Office Action acknowledges that Hinks does not disclose the above-recited substituting operation. But the Office Action argues that Sun makes up for this deficiency through its various gettext()-related routines. The Office Action then acknowledges that Sun itself is deficient because its gettext() instructions do not act as substitutes in the place of *removed* content. But the Office Action states that Lyapustina makes up for this deficiency by disclosing substituting content with a unique identifier string.

In replying to this rejection, the Applicant hereby incorporates all of the arguments presented in the previous Response, filed on April 28, 2006. In view of the

After-Final status of the present case, the Applicant addresses the outstanding arguments in a more succinct form hereinbelow.

Hinks describes an Export/Import module for extracting strings and translatable information from an application program. The extracted information is stored in a translation table. Hinks uses various editors to transform the extracted information. Hinks then relies on the Export/Import module to merge the translated information back into the application product to provide a compiled product with translated resources. See generally column 3, lines 1-43 of Hinks.

As a first deficiency of the rejection, Hinks does not disclose the elements for which it was relied upon. For example, Hinks does not disclose at least the operation of "at runtime, retrieving the compiled document and populating the compiled document with a desired version of the original locale-sensitive content." In Hinks, the Export/Import module does not re-insert the translated information into an application product at runtime, but, rather, at compile time. This is made clear at various junctures in Hinks, such as column 3, lines 36-39 ("The file may be simply stored back with the source files, as a translated resource files(s); or, the translated resource file may be compiled and bound back into the target program directly"). Also note column 8, lines 24-37. It is clear from these passages that Hinks' tools are being used to modify code during the compilation stage, not to dynamically modify code at runtime.

As a second deficiency of the rejection, one of ordinary skill in the art would not have been motivated to combine Hinks, Sun, and Lyapustina in the manner set forth in the rejection. In general, the references are not combinable because the references seek to modify code using incompatible paradigms.

Consider first the approach taken by Hinks and its underlying motivations. In discussing various deficiencies of other approaches, Hinks states that "Merely separating

text in a user interface from one's program is not an acceptable solution, however" (column 1, lines 30-32). Hinks further states that, "At the level of application development, software translation is typically accomplished by extracting 'translatable strings' from various sources, translating them into the local language, and reinserting them back into the original sources for recompilation and linking. The approach has distinct disadvantages, however" (column 2, lines 30-35). Hinks' solution attempts to overcome these perceived drawbacks by first extracting a wealth of information from original sources (including information which supplements the translatable strings, such as position information). Hinks then uses a suite of standardized editing tools to manipulate the extracted information. This presumably allows a human translator to exercise more nuanced and accurate control over the translation of various types of program content.

Sun uses an entirely different approach based on different underlying motivations. Sun provides tools that allow a programmer, *while writing a program*, to mark certain translatable content with gettext() instructions. After the program is written in such a manner, Sun allows the user to run a utility on the program to extract the marked translatable content for storage in a portable object file. The content in the portable object file can be translated into a different language and then the file can be converted into a message object file. The gettext() instructions in the program can then be used to retrieve the translations in the message object file.

Considering the widely differing approaches used by Hinks and Sun, one having ordinary skill in the art would not have looked to Sun to supplement Hinks. To summarize, Hinks' basic approach is to empower the human translator with advanced editing tools, without adopting any simplifying constraints regarding the form of the source code. In fact, Hinks emphasizes the fact that its tools are standardized and can be

applied to various code products (e.g., see column 3, lines 44-55). In contrast, Sun delegates some of the decision-making pertaining to internationalization to the programmer, that is, by requiring the programmer to integrate gettext() instructions at appropriate locations in the source code when writing the code. The fact that the code is marked in advanced serves as a simplifying assumption, allowing the actual process of extraction, translation, and reinsertion to proceed in a more straightforward and "mechanical" manner than Hinks. The differences between Hinks and Sun reflect a significant divergence in these systems' foundational principles, further indicating that it would not have been obvious to supplement Hinks by using the gettext()-related routines disclosed by Sun.

The above conclusion becomes ever clearer in view of the fact that, in columns 1 and 2, Hinks can be seen to disparage techniques that are similar in one or more respects to Sun. For instance, to repeat, Hinks states that "Merely separating text in a user interface from one's program is not an acceptable solution, however" (column 1, lines 30-32). Insofar as Sun maintains message object files that contain displayable content, Hinks can be said to be teaching away from the Sun reference.

Adding Lyapustina to the combination of references does not cure the above incongruity, and, in fact, compounds it. Lyapustina describes a technique for automatically substituting unique macro strings for original strings within code. At compilation time, a compiler may substitute each instance of the unique macro string with its associated identified string. See column 4, lines 7-31. First, consider the combination of Hinks and Lyapustina. Lyapustina is like Sun in that it represents an attempt to simplify the translation process through an automatic extraction and reinsertion paradigm. As pointed out above, the introductory portions of Hinks state that these kinds of approaches have various disadvantages, indicating that if one having

ordinary skill in the art wanted to modify Hinks' approach, he or she would not move in the direction of simplified and streamlined automation.

Second, consider the combination of Sun and Lyapustina. Sun describes the use of gettext()-related routines for retrieving translated text (e.g., note page 98 of Sun). In contrast, as mentioned above, Lyapustina describes the use of unique macro strings in a program file which are used at *compile time* to perform string substitution, meaning that, at *runtime*, no substitution is performed. Sun and Lyapustina therefore describe two different techniques used in very different contexts. Hence, there would be no motivation to use the substitution mechanism described by Lyapustina in Sun's technique (even if complemented by Hinks). Namely, if Lypustina's technique is used (in which strings are substituted at compile time), there is no longer any need for function calls which can be invoked at runtime. This means that these two techniques do not complement each other; rather, these techniques are mutually exclusively, such that if one technique is used, the other is not needed.

Finally, the incongruities associated with the proposed combination of any two of the applied references remain intact when all three references are combined together.

The Final Office Action provides various arguments which are relevant to points raised above. In page 4, last paragraph of the Final Office Action, the Patent Office argues, in essence, that Hinks' introductory remarks focus on character-based translation tools. The Patent Office points out that Sun is more advanced in that it allows a user to adjust the size and position of translatable elements. The Office Action concludes that this disclosure indicates that Hinks does not teach away from Sun and Lyapustina; in fact, the Office Action states that this teaching provides further motivation to combine the references. It is true that Sun discloses supplemental tools to address size and position issues, but this does not diminish the otherwise significant differences in approach among

Hinks, Sun, and Lyapustina. To repeat, for instance, Hinks requires the programmer to precondition the code so that it can be easily manipulated using specialized gettext()-related routines, while Hinks makes no such assumptions, relying on the work of a human translator in conjunction with specialized editing tools to modify the program code in an appropriate manner. Moreover, it is pointed out that the question before us is focused to whether it is obvious to use Sun's gettext() calls in Hinks system, rather than more defuse question of whether *any aspect* of Sun's system could be used in Hinks' system.

On page 5, first full paragraph, the Office Action addresses the Applicant's assertion that Lyapustina operates at compile time. The Office Actions states, "However, both references are concerned with the coded representation of strings in source code which exists independently from a particular compile-time or run-time." To the extent understood, the Applicant disagrees with this statement. A pertinent element of claim 20 requires that the "retrieving" and "populating" be performed at runtime. The reinsertion of code in Lyapustina is performed at compile time, not at runtime, and thus, this aspect of Lyapustina is considered relevant for the reasons set forth above. The Office Action continues by stating, "Further, Applicant's argument does not diminish the teaching of Lyapustina, that the hard coding of strings is disadvantageous and inflexible (see Lyapustina column 1 lines 64-65)." The Patent Office relies on this portion to Lyapustina to establish a motivation to combine Sun with Lyapustina (as set forth in page 9 of the Office Action). However, Lypustina does not provide a general directive to avoid hard coding, but merely describes a technique for translating hard coded content that happens to use macro strings (note column 2, lines 45-47). Further, Lypustina does not supply motivation to replace Sun's gettext() instructions with macro strings because the gettext() instructions are foundational building blocks of Sun's technique; that is, to remove these instructions would undermine the basic nature of Sun's technique.

For at least the above-stated reasons, there is no motivation to combine Hinks, Sun, and Lyapustina. As stated in MPEP § 2143.01, if the proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984). Further, if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious. *In re Ratti*, 270 F.2d 810, 123 USPQ 349 (CCPA 1959). The Applicant submits that adding Sun and/or Lyapustina to Hinks would run counter to the objectives of Hinks set forth in columns 1 and 2 of Hinks. Further, adding Lyapustina's substitution technique to Sun would entirely eviscerate the foundational principles of the gettext() routines described in Sun. As such, these kinds of contradictory combinations are specifically proscribed by the MPEP and relevant case law.

For completeness, the Applicant submits that the Gao and Besaw references do not make up for the above-identified deficiencies of Hinks, Sun, and Lyapustina, regardless of how these documents are combined.

For at least all of the above-stated reasons, the Applicant submits that none of the applied documents renders independent claim 20 obvious, whether these documents are considered individually or in any combination. The other pending independent claims (i.e., claims 32, 49, and 55) recite related features to claim 20, and are therefore allowable for reasons that are similar to those presented for claim 20. The dependent claims are allowable at least by virtue of their dependency on their respective independent claims. In addition, the dependent claims add subject matter which further distinguishes the invention recited in these claims over the applied art.

For at least the above-stated reasons, the Applicant respectfully requests the withdrawal of the 35 U.S.C. § 103(a) rejections.

Regarding the Appropriateness of Entry of Response under 37 C.F.R. § 1.116

37 C.F.R. § 1.116 reads, in part, that, an After Final Amendment "may be made canceling claims or complying with any requirement of form expressly set forth in a previous Office action." The present Response responds to the objections to the drawing and the claims (which are essentially matters of form) in a straightforward manner. For this reason, Applicant submits that entry of the present Response is appropriate under 37 C.F.R. § 1.116, and respectfully requests such entry.

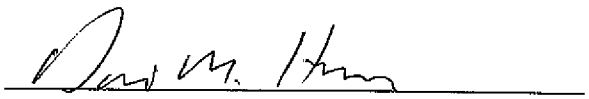
Conclusion

The arguments presented above are not exhaustive; Applicant reserves the right to present additional arguments to fortify its position. Further, Applicant reserves the right to challenge the prior art status of one or more documents cited in the Office Action.

All objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the present application is in condition for allowance and such allowance is respectfully solicited. The Examiner is urged to contact the undersigned if any issues remain unresolved by this Amendment.

Respectfully Submitted,

Dated: 11/20/2006

By: 

David M. Huntley
Reg. No. 40,309
(509) 324-9256